LA-UR- 04-4561

Approved for public release; distribution is unlimited.

Title: UPDATING MODEL PARAMETERS USING BAYESIAN NETWORKS

Author(s): Christine A. Treml
Timothy J. Ross

Submitted to: PMC 2004
9th ASCE/EMD/SEI/GI/AD Joint Speciality Conference on Probabilistic Mechanics and Structural Reliability

MODEL PARAMETER UPDATING USING BAYESIAN NETWORKS

C. Treml[a] and **T. Ross[b]**

[a]Los Alamos National Laboratory
TA-33, MS J570
Los Alamos, NM 87545
treml@lanl.gov

[b]Department of Civil Engineering
University of New Mexico
Albuquerque, NM 87131
ross@unm.edu

This paper outlines a model parameter updating technique for a new method of model validation using a modified model reference adaptive control (MRAC) framework with Bayesian Networks (BNs). The model parameter updating within this method is generic in the sense that the model/simulation to be validated is treated as a black box. It must have updateable parameters to which its outputs are sensitive, and those outputs must have metrics that can be compared to that of the model reference, i.e., experimental data. Furthermore, no assumptions are made about the statistics of the model parameter uncertainty, only upper and lower bounds need to be specified.

This method is designed for situations where a model is not intended to predict a complete point-by-point time domain description of the item/system behavior; rather, there are specific points, features, or events of interest that need to be predicted. These specific points are compared to the model reference derived from actual experimental data. The logic for updating the model parameters to match the model reference is formed via a BN. The nodes of this BN consist of updateable model input parameters and the specific output values or features of interest. Each time the model is executed, the input/output pairs are used to adapt the conditional probabilities of the BN. Each iteration further refines the inferred model parameters to produce the desired model output. After parameter updating is complete and model inputs are inferred, reliabilities for the model output are supplied. Finally, this method is applied to a simulation of a resonance control cooling system for a prototype coupled cavity linac. The results are compared to experimental data.

MODEL PARAMETER UPDATING USING BAYESIAN NETWORKS

C.A. Treml
Los Alamos National Laboratory, Los Alamos, NM 87545
treml@lanl.gov

T.J. Ross
University of New Mexico, Albuquerque, NM 87131
ross@unm.edu

Abstract

This paper outlines a model parameter updating technique for a new method of model validation using a modified model reference adaptive control (MRAC) framework with Bayesian Networks (BN). This method is designed for situations where a model is not intended to predict a complete point-by-point time domain description of the item/system behavior; rather, there are specific points, features, or events of interest that need to be predicted. These specific points are compared to the model reference derived from actual experimental data. The logic for updating the model parameters to match the model reference is formed via a BN. The nodes of this BN consist of updateable model input parameters and the specific output values or features of interest. Each time the model is executed, the input/output pairs are used to adapt the conditional probabilities of the BN. Each iteration further refines the inferred model parameters to eventually produce the desired model output. This method is applied to a simulation of a resonance control cooling system for a prototype coupled cavity linac. The results are compared to experimental data.

Introduction

The block diagram in Figure 1 illustrates the model parameter updating within a new method by (Treml 2004) for model validation using a modified model reference adaptive control (MRAC) framework with Bayesian "Belief" Networks (BNs or BBNs). BBNs (as in Figures 2 and 3) are acyclic graphical networks representing casual relationships

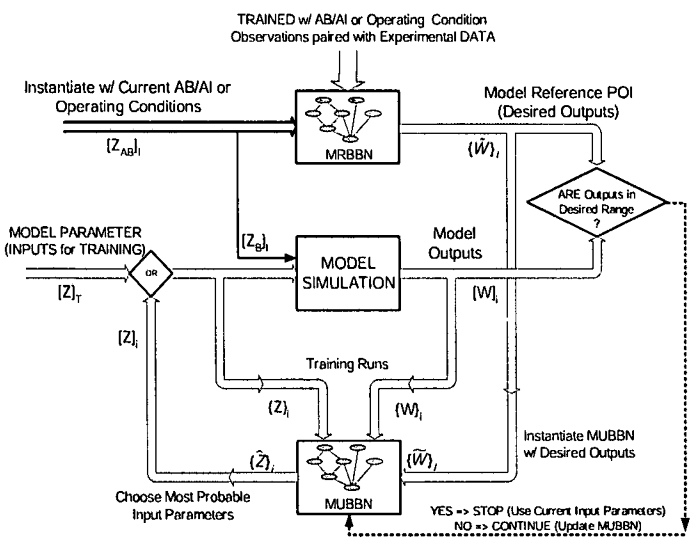

Figure 1. Block Diagram of Model Parameter Updating Method

among variables. The directed acyclic graph represents qualitative dependencies among random variables, and the conditional probabilities quantify these dependencies. Within the graph, nodes represent the modeled variables and directed edges (arrows) represent the causal (conditional) relationships between the modeled variables. If information is known about a variable, it is referred to as evidence and evidence is introduced to the BBN by instantiating that node with the evidence. The evidence is then propagated throughout the BBN, inferring new probabilities about the remaining nodes/variables (Pearl 1988).

The model parameter updating within the MRAC/BBN method is generic, in that the model/simulation to be validated is treated as a black box. This implies that any type of model/simulation can be validated using this procedure provided it has updateable parameters to which its outputs are sensitive and that those outputs have metrics that can be compared to that of the model reference, *i.e.*, experimental data. The black box model can be of a single item, *i.e.* a computer chip, beam, tire, bearing, or spherical steel shell, or a complex system, *i.e.* a chemical process, car, civil structure, or cooling system.

The most probable model inputs are inferred when the model updating BBN (MUBBN) is instantiated with desired outputs, *i.e.* the model reference. The model reference is in the form of specific points, features, or events of interest that need to be predicted, such as point of failure, rise/settling times, a steady-state value, or a particular statistic. These predictions or points of interest (P.O.I.) are the basis for the validation metrics. The MRBBN must infer them, the model to be validated must calculate them, and the P.O.I. must be measured or observed in the validation experiments (Treml 2000). These inferred inputs are fed into the computer model, and the corresponding outputs are then compared to the model reference. If the model outputs are within a pre-described tolerance of the model reference, then the process or parameter updating loop is stopped. At this point, the model is validated for the set of conditions from which the model reference was inferred.

Structure and Training of MUBBN

The logic for the parameter updating process is provided by the MUBBN. The MUBBN is composed of discrete nodes that represent the P.O.I. of the model/experiment, the uncertain updateable model parameters, and qualitative nodes. The general form of the MUBBN is shown in Figure 2. The qualitative nodes can represent any subjective or qualitative aspects of the model response or experimental outcomes that the modeler has observed to be useful in the predictive process. A spread of model runs are made to obtain model input/output pairs used to train the conditional probabilities for the MUBBN. These conditional probabilities are generated using the EM (expectation maximization) algorithm developed by Lauritzen (1995) in conjunction with the MUBBN node/edge structure and the table of model input/output set pairs. A key point of this step is that no assumptions are made about the statistics of the model parameter uncertainty. An acceptable range for each variable must be made, but nothing more. This is a departure from many model validation techniques, where the tendency is to arbitrarily assume that the parameter uncertainty is Gaussian, then base the validation upon that assumption.

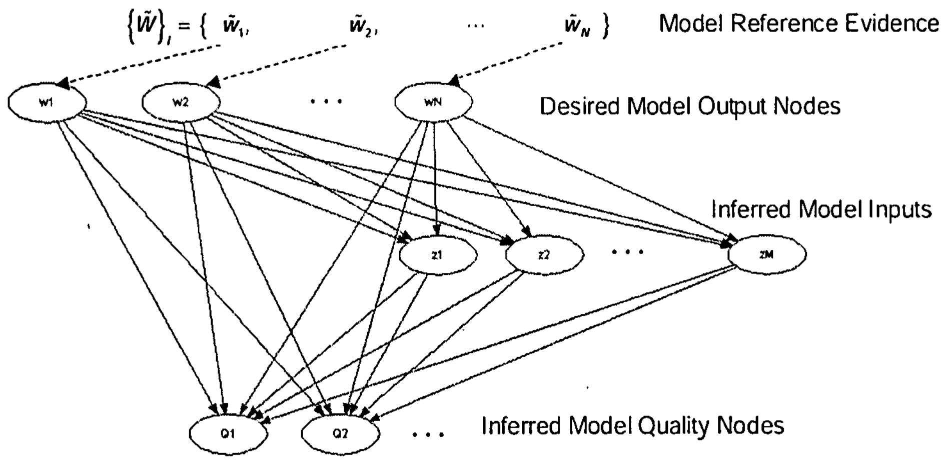

Figure 2. General Model Reference BBN (MUBBN)

The training runs for the MUBBN are like a Monte-Carlo process, all possible continuous combinations of model inputs are not represented, hence, the need for an iterative process. Each time the model output P.O.I. are unsuccessfully compared to the model reference, the inferred model parameter input bins are "re-gridded" in a specified manner, the model is run, and more model input/output pairs are generated. These input/output pairs are used to refine or adapt the conditional probabilities within the MUBBN. Eventually this learning refinement converges to a solution, provided an input parameter combination exists that can produce the desired model outputs within the specified tolerances. After model inputs are inferred that yield a set of model outputs within the specified tolerances, the bin widths and inferred probabilities of these input/output vectors can be used to make statements about the reliability of the model predictions.

Parameter Updating Algorithm

Once trained, the model output nodes of the MUBBN are instantiated with the model reference or desired outputs. For each iteration of parameter updating, the proposed input set $[Z]_i$ is used to generate the model output $[W]_i$ for testing against the model reference $\{\tilde{W}\}_i$. Each time this is done the model input/output variable pair $[Z]_i, [W]_i$ is introduced as the bin evidence pair $\{Z\}_i, \{W\}_i$ to the MUBBN and used to update or adapt its probabilities. Probability adaptation also takes place when grid-refinement runs are made. The MUBBN nodes $(W1, W2, \cdots WN)$ are divided into discrete bins $\langle w_j \rangle^k$ where: $\langle w_j \rangle^k = \left\{ w_j : (w_j)_{Lk} \le w_j < (w_j)_{Uk} \right\}$ is the k^{th} bin of the node Wj from Figure 2. If $\langle \tilde{w}_i \rangle \in \langle w_j \rangle^k$, then $\langle w_j \rangle^j$ is instantiated. This bin evidence $\langle \tilde{w}_i \rangle$ is then propagated through the MUBBN, and probabilities are inferred for each of the node states for all the input parameter nodes $(Z1, Z2, \cdots ZM)$. This inferred input information $\{\hat{Z}\}_i$ is indicated by the "^" symbol and written: $\{\hat{Z}\}_i = \{[\hat{Z}_1],[\hat{Z}_2],\cdots[\hat{Z}_M]\}_i$, where the bin set $[\hat{Z}_j]$ corresponds to the updateable model input parameter, z_j. Furthermore, the bin set and accompanying probabilities or inferred beliefs are written:

$$[\hat{Z}_j]_i = \begin{bmatrix} \langle \hat{z}_j \rangle^1 & , prob\left(\langle \hat{z}_j \rangle^1 \mid \{\tilde{W}\}_l\right) \\ \langle \hat{z}_j \rangle^2 & , prob\left(\langle \hat{z}_j \rangle^2 \mid \{\tilde{W}\}_l\right) \\ & \vdots \\ \langle \hat{z}_j \rangle^{mj} & , prob\left(\langle \hat{z}_j \rangle^{mj} \mid \{\tilde{W}\}_l\right) \end{bmatrix}_i .$$

where mj is the number of bins or node states for the input parameter z_j, and i implies the i^{th} iteration of $\{\hat{Z}\}_i$. The output bin $\langle \hat{z}_j \rangle^k$ is the k^{th} bin for z_j, i.e. $\langle \hat{z}_j \rangle^k = \{z_j : (z_j)_{Lk} \le z_j < (z_j)_{Uk}\}$, where Uk and Lk indicate the upper and lower bounds of the bin k, respectively. The overall range of z_j is $\{z_j : (z_j)_{L1} \le z_j < (z_j)_{Umj}\}$. Once the most probable model input bins $[\langle z_1 \rangle^*, \langle z_2 \rangle^*, \cdots \langle z_M \rangle^*]$ are determined, $\langle z_j \rangle^* = \left[\langle z_j \rangle^k : \exists \max\left\{prob\left(\langle z_j \rangle^k \mid \{\tilde{W}\}_l\right)\right\}; \quad k = 1, 2, \cdots mj\right]$, this information is converted to a single set of model input values, $[Z]_i = [z_1, z_2, \cdots z_M]_i$. Each value z_j^* is chosen as the midpoint of the corresponding most probable input bin interval $\langle z_j \rangle^*$ to form the input set $[Z]_i = [z_1^*, z_2^*, \cdots z_M^*]_i$. Next, the simulation is run with the updateable model parameters set to these values, and the outputs $[W]_i$ are generated. The model outputs $[W]_i$ are then tested to determine whether they are within the bin or tolerance of the model reference $\{\tilde{W}\}_l$: $w_k \in \langle \tilde{w}_k \rangle_l = \{\tilde{w}_k : [(\tilde{w}_k)_{L1} < \tilde{w}_k \le (\tilde{w}_k)_{U1}]\} \; \forall k$?
If this test is true, then the iteration is stopped provided the input parameter grid has been refined to reflect the model input parameter uncertainty. If the test is not true, then the MUBBN is updated or adapted and iteration is continued.

Application of MRAC/BBN Algorithm

MRAC/BBN method was applied to a model of a resonance control cooling system (RCCS) for a coupled cavity linac (CCL). For a CCL, resonant frequency is primarily a function of the geometry of the copper cavities and couplers. As RF power is dissipated in the cavity walls, the copper will heat, then expand, and its resonant frequency will decrease. This frequency shift is controlled by the closed loop RCCS. To apply the parameter updating scheme, one set of experimental data (shown in Figure 4) was used as the model reference P.O.I. for a specific set of operating conditions. The P.O.I. values

T_lo = 29.95°C, T_hi = 29.20°C, and t_Peak = 90 sec. shown in Figure 4 are the steady-state surface temperature of the cavity at high and low power, and the time at which the surface temperature peaks after a power loss, respectively. The bin widths (tolerances) and ranges for the reference P.O.I. are shown in Figure 3. The corresponding, updateable model input parameters are $[Z]_i = [hA\_lo, hA\_hi, k]_i$, that are heat transfer coefficients at low and high power and a term effected by the amount of cooper heated, respectively. Their desired uncertainty bounds were chosen at ~5% of their anticipated final values: $\Delta\langle hA\_lo\rangle = 50,\ \Delta\langle hA\_hi\rangle = 500,\ \textit{and}\ \Delta\langle k\rangle = 2000$. A MUBBN for the RCCS model was formed using the aforementioned model inputs, model reference P.O.I., and a model quality node. A working HUGIN (1999) depiction of the RCCS MUBBN is shown in Figure 3, along with the parameter bins and probabilities for each input and output variable and the model quality node.

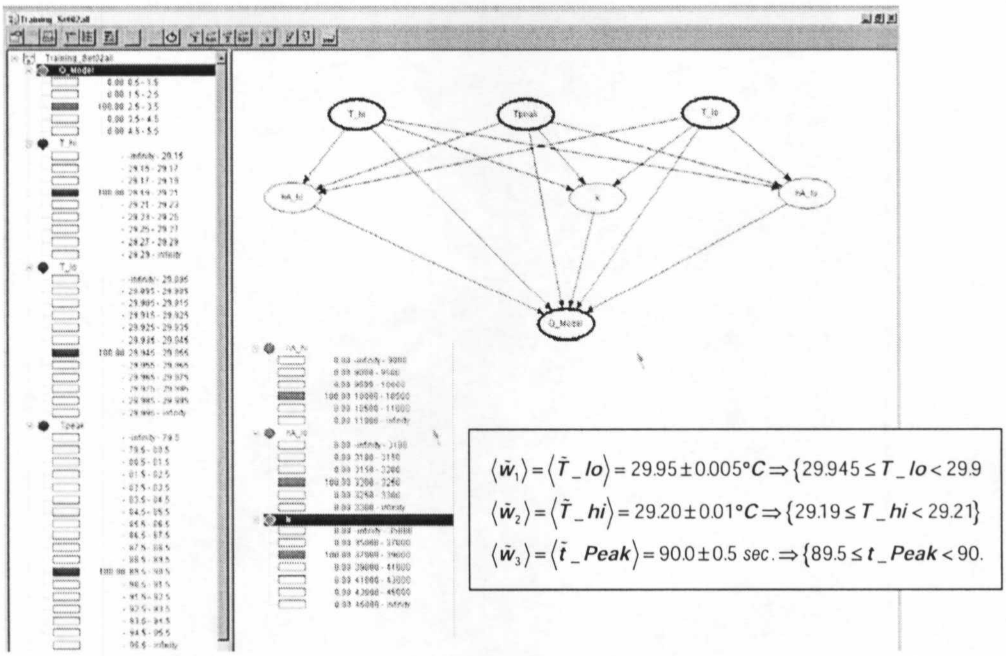

Figure 3. MUBBN for RCCS Prototype CCL Model Validation

Initially, the unvalidated model deviated dramatically from the experimental data, as illustrated it Figure 4. After four complete iteration steps consisting of approximately 350 model runs, parameter updating was successful in finding an acceptable set of model input parameters to produce a model output response within P.O.I. tolerances. The most probable values for the uncertain model input parameters are: $hA\_lo = 3{,}225 \pm 25$, $hA\_hi = 10{,}250 \pm 250$, and $k = 38{,}000 \pm 1000$. Given these inputs with their specified uncertainty bounds, the model output will be $29.19 \le T\_hi < 29.21°C$ (with 87.3% reliability), $29.935 \le T\_lo < 29.955°C$ (with 98.6% reliability) and $78 \le t\_Peak < 93\,sec$. The T_hi and T_lo P.O.I. matching can be considered a success from an engineer's standpoint. However, t_Peak indicates that the tolerance on t_Peak is not obtainable given the current model parameter uncertainty.

Despite the fact that updated input parameters now yield a model output response that matches all specified P.O.I. within tolerance, obviously, there remains some modeling error that can be corrected, or in this case, some un-modeled dynamics that can be added.

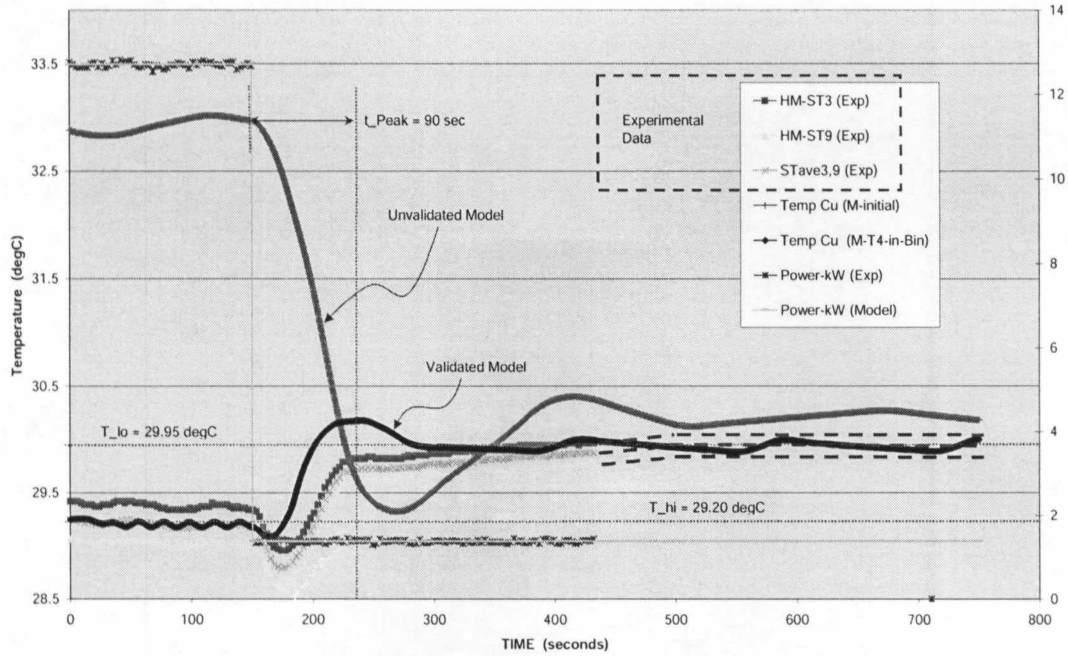

Figure 4. Results of Parameter Updating Compared with Experimental Data

Conclusions

As applied to the RCCS simulation, the MRAC/BBN validation method functioned successfully. The parameter updating algorithm produced a set of input parameters with a specified uncertainty bound that resulted in a model output response whose P.O.I. were within model reference tolerances. Also, reliabilities for the specified tolerances of model outputs given the input uncertainty bounds were inferred by the MUBBN.

References

HUGIN (1999), Hugin Expert A/S, User's Manual/Version 5.3 (build 802).

Lauritzen, S.L. (1995), "The EM algorithm for graphical association models with missing data," Computational Statistics & Data Analysis, vol. 19, pp. 191-201.

Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, INC.

Treml, C.A. (2000), "Using Bayesian Belief Networks and an As-Built/As-Is Assessment for Quantifying and reducing Uncertainty in Prediction of Buckling Characteristics of Spherical Shells," *8th Specialty Conference on Probabilistic Mechanics and Structural Reliability*, Notre Dame, Indiana, July 24-26, 2000, Proceedings, no.(170), pp. 1-6.

Treml, C.A. (2004), "Model Validation Using Bayesian Belief Networks," Los Alamos National Laboratory Report, LA-14140-T, Los Alamos, NM.